Foundations of Artificial Intelligence

F5. Automated Planning: Abstraction

Malte Helmert

University of Basel

May 7, 2025

Automated Planning: Overview

Chapter overview: automated planning

- F1. Introduction
- F2. Planning Formalisms
- F3. Delete Relaxation
- F4. Delete Relaxation Heuristics
- F5. Abstraction
- F6. Abstraction Heuristics

Planning Heuristics

We consider two basic ideas for general heuristics:

- Delete Relaxation
- Abstraction $\rightsquigarrow$ this chapter

Abstraction: Idea

Estimate solution costs by considering a smaller planning task.

F5.1 SAS^+

SAS^+ Encoding

- in this chapter: SAS^+ encoding instead of STRIPS (see Chapter F2)
- difference: state variables v not binary, but with finite domain $\mathrm{dom}(v)$
- accordingly, preconditions, effects, goals specified as partial assignments
- everything else equal to STRIPS

(In practice, planning systems convert automatically between STRIPS and SAS^+.)

SAS$^+$ Planning Task

Definition (SAS$^+$ planning task)

A SAS$^+$ planning task is a 5-tuple $\Pi = \langle V, \mathrm{dom}, I, G, A \rangle$ with the following components:

- V: finite set of state variables
- dom: domain; $\mathrm{dom}(v)$ finite and non-empty for all $v \in V$
  - states: total assignments for V according to dom
- I: the initial state (state = total assignment)
- G: goals (partial assignment)
- A: finite set of actions a with
  - $\mathit{pre}(a)$: its preconditions (partial assignment)
  - $\mathit{eff}(a)$: its effects (partial assignment)
  - $\mathit{cost}(a) \in \mathbb{N}_0$: its cost

German: SAS$^+$-Planungsaufgabe

State Space of SAS$^+$ Planning Task

Definition (state space induced by SAS$^+$ planning task)

Let $\Pi = \langle V, \mathrm{dom}, I, G, A\rangle$ be a SAS$^+$ planning task.
Then Π induces the state space $\mathcal{S}(\Pi) = \langle S, A, cost, T, s_{\mathrm{I}}, S_{\mathrm{G}}\rangle$:

- set of states: total assignments of V according to dom
- actions: actions A defined as in Π
- action costs: $cost$ as defined in Π
- transitions: $s \xrightarrow{a} s'$ for states s, s' and action a iff
  - $pre(a)$ agrees with s (precondition satisfied)
  - s' agrees with $eff(a)$ for all variables mentioned in eff; agrees with s for all other variables (effects are applied)
- initial state: $s_{\mathrm{I}} = I$
- goal states: $s \in S_{\mathrm{G}}$ for state s iff G agrees with s

German: durch SAS$^+$-Planungsaufgabe induzierter Zustandsraum

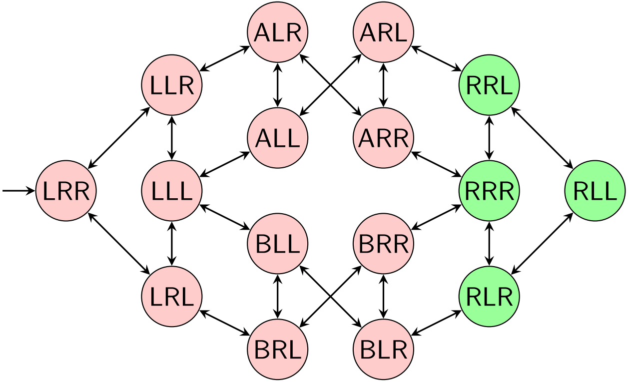
Example: Logistics Task with One Package, Two Trucks

Example (one package, two trucks)

Consider the SAS$^+$ planning task $\langle V, \mathrm{dom}, I, G, A\rangle$ with:

- $V = \{p, t_\mathrm{A}, t_\mathrm{B}\}$
- $\mathrm{dom}(p) = \{\mathrm{L}, \mathrm{R}, \mathrm{A}, \mathrm{B}\}$ and $\mathrm{dom}(t_\mathrm{A}) = \mathrm{dom}(t_\mathrm{B}) = \{\mathrm{L}, \mathrm{R}\}$
- $I = \{p \mapsto \mathrm{L}, t_\mathrm{A} \mapsto \mathrm{R}, t_\mathrm{B} \mapsto \mathrm{R}\}$
- $G = \{p \mapsto \mathrm{R}\}$
- $A = \{load_{i,j} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{\mathrm{L}, \mathrm{R}\}\}$
  $\cup \{unload_{i,j} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{\mathrm{L}, \mathrm{R}\}\}$
  $\cup \{move_{i,j,j'} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j, j' \in \{\mathrm{L}, \mathrm{R}\}, j \neq j'\}$ with:
  - $load_{i,j}$ has preconditions $\{t_i \mapsto j, p \mapsto j\}$, effects $\{p \mapsto i\}$
  - $unload_{i,j}$ has preconditions $\{t_i \mapsto j, p \mapsto i\}$, effects $\{p \mapsto j\}$
  - $move_{i,j,j'}$ has preconditions $\{t_i \mapsto j\}$, effects $\{t_i \mapsto j'\}$
  - All actions have cost 1.

State Space for Example Task



- state $\{p \mapsto i, t_{\mathrm{A}} \mapsto j, t_{\mathrm{B}} \mapsto k\}$ denoted as ijk
- annotations of edges not shown for simplicity
- for example, edge from LLL to ALL has annotation $load_{\mathrm{A,L}}$

F5.2 Abstractions

State Space Abstraction

State space abstractions drop distinctions between certain states, but preserve the state space behavior as well as possible.

- An abstraction of a state space $\mathcal{S}$ is defined by an abstraction function α that determines which states can be distinguished in the abstraction.
- Based on $\mathcal{S}$ and α, we compute the abstract state space $\mathcal{S}^\alpha$ which is "similar" to $\mathcal{S}$ but smaller.
- main idea: use optimal solution cost in $\mathcal{S}^\alpha$ as heuristic

German: Abstraktionsfunktion, abstrakter Zustandsraum

Induced Abstraction

Definition (induced abstraction)

Let $\mathcal{S} = \langle S, A, cost, T, s_\mathsf{I}, S_\mathsf{G} \rangle$ be a state space, and let $\alpha : S \to S'$ be a surjective function.

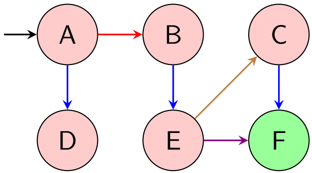
The abstraction of $\mathcal{S}$ induced by α, denoted as $\mathcal{S}^\alpha$, is the state space $\mathcal{S}^\alpha = \langle S', A, cost, T', s'_\mathsf{I}, S'_\mathsf{G} \rangle$ with:

- $T' = \{\langle \alpha(s), a, \alpha(t) \rangle \mid \langle s, a, t \rangle \in T\}$
- $s'_\mathsf{I} = \alpha(s_\mathsf{I})$
- $S'_\mathsf{G} = \{\alpha(s) \mid s \in S_\mathsf{G}\}$

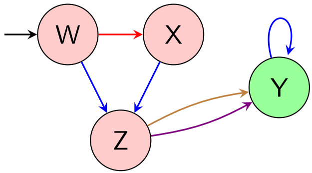
German: induzierte Abstraktion

Abstraction: Example

concrete state space with
states $S = \{A, B, C, D, E, F\}$



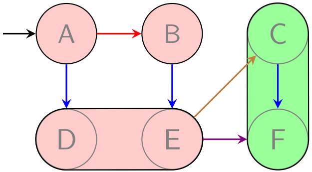
abstract state space with
states $S^\alpha = \{W, X, Y, Z\}$



abstraction function $\alpha : S \to S^\alpha$

$\alpha(A) = W \quad \alpha(B) = X \quad \alpha(C) = Y$
$\alpha(D) = Z \quad \alpha(E) = Z \quad \alpha(F) = Y$

intuition: grouping states



F5.3 Summary

Summary

- basic idea of abstractions: simplify state space by considering a smaller version
- formally: abstraction function α maps states to abstract states and thus defines which states can be distinguished by the resulting abstraction
- induces abstract state space